

1 Claims 1, 9, 10, 11, 15, 16, 23, 33, 41, 42, 43 and 44 are amended.

2 Claims 1, 3-11, 13-19, 21-26, 28-34 and 36-52 are pending and are listed as
3 follows:

4

5 **1. (Currently Amended)** One or more computer storage media
6 comprising a flash memory driver that is executable by a computer to interface
7 between a file system and one or more flash memory media, the flash memory
8 driver comprising:

9 flash abstraction logic that is invokable by the file system to manage flash
10 memory operations without regard to the type of the one or more flash memory
11 media; and

12 flash media logic configured to interact with different types of the flash
13 memory media, wherein the flash media logic is programmable to permit users to
14 match particular medium requirements of a specific manufacturer;

15 wherein the flash abstraction logic invokes the flash media logic to perform
16 memory operations that are potentially performed in different ways depending on
17 the type of the flash memory media, and further wherein the flash memory driver
18 is flash memory medium agnostic, and wherein one of the flash memory
19 operations includes performing wear-leveling operations associated with the flash
20 memory medium by way of circular and continuous advancement of a write
21 pointer, and wherein the flash memory driver resides as a component within an
22 operating system of the computer.

23

24 **2. (Cancelled).**

25

1 3. **(Original)** The flash memory driver as recited in Claim 1,
2 wherein one of the flash memory operations includes maintaining data integrity of
3 the flash memory medium.

4

5 4. **(Original)** The flash memory driver as recited in Claim 1,
6 wherein one of the flash memory operations includes handling recovery of data
7 associated with the flash memory medium after a power-failure.

8

9 5. **(Original)** The flash memory driver as recited in Claim 1,
10 wherein one of the flash memory operations includes mapping status information
11 associated with physical sectors of the flash memory medium for use by the file
12 system.

13

14 6. **(Previously Presented)** The flash memory driver as recited in
15 Claim 1, wherein the flash medium logic is further configured to translate
16 commands received from the file system to physical sector commands for issuance
17 to the flash memory media.

18

19 7. **(Previously Presented)** The flash memory driver as recited in
20 Claim 1, wherein the flash medium logic is user programmable to read, write and
21 erase data to and from the flash memory media.

22

23 8. **(Previously Presented)** The flash memory driver as recited in
24 Claim 1, wherein the flash media logic is configured to perform error code
25 correction associated with the flash memory media.

1 **9. (Currently Amended)** A flash driver, comprising:
2 flash abstraction logic, interposed between a file system and a flash
3 memory medium, configured to:

4 (a) map a logical sector status from the file system to a physical sector
5 status of the flash memory medium; and

6 (b) maintain memory requirements associated with operating the flash
7 memory medium;

8 wherein the flash driver is located remote from the flash memory medium,
9 and wherein the memory requirements include managing wear-leveling operations
10 associated with the flash memory medium by way of circular and continuous
11 advancement of a write pointer, and wherein the flash driver resides as a
12 component within an operating system of a computer;

13 flash medium logic that is programmable to permit users to match
14 particular medium requirements of a specific manufacturer.

15
16 **10. (Currently Amended)** The flash driver as recited in Claim 9,
17 ~~further comprising a wherein the user~~ programmable flash medium logic [[,]] is
18 configured to read, write and erase data to and from the flash memory medium.

19
20 **11. (Currently Amended)** The flash driver as recited in Claim 9,
21 ~~further comprising a wherein the user~~ programmable flash medium logic [[,]] is
22 configured to receive and translate specific operational commands from the file
23 system associated with reading and writing data to the flash memory medium.

24
25 **12. (Cancelled).**

1
2 **13. (Original)** The flash driver as recited in Claim 9, wherein the
3 memory requirements include maintaining data integrity of the flash memory
4 medium.

5
6 **14. (Original)** The flash driver as recited in Claim 9, wherein the
7 memory requirements include handling recovery of data associated with flash
8 memory medium after a power-failure.

9
10 **15. (Currently Amended)** The flash driver as recited in Claim 9,
11 ~~further comprising a~~ wherein the flash medium logic is [[,]] programmably
12 configurable by a user to perform error code correction associated with the flash
13 memory medium.

14
15 **16. (Currently Amended)** A flash driver, comprising:
16 user programmable flash medium logic, configured to read, write and erase
17 data to and from a flash memory medium, wherein the flash medium logic is
18 programmable to permit users to match particular flash medium requirements of a
19 specific manufacturer; and

20 flash abstraction logic, interposed between a file system and flash memory
21 medium to maintain universal requirements for the operation of the flash memory
22 medium;

23 wherein the flash memory driver is flash memory medium agnostic, and
24 wherein the universal requirements include managing wear-leveling operations
25 associated with the flash memory medium by way of circular and continuous

1 advancement of a write pointer, and wherein the flash driver is defined as a
2 component within an application.

3

4 **17. (Original)** The flash driver as recited in Claim 16, wherein the
5 flash abstraction logic passes specific commands associated with certain types of
6 flash memory media directly to the flash medium logic for translation and
7 execution.

8

9 **18. (Original)** The flash driver as recited in Claim 16, wherein the
10 flash abstraction logic is an interface between the flash medium logic and the file
11 system.

12

13 **19. (Original)** The flash driver as recited in Claim 16, wherein the
14 universal requirements include maintaining data integrity of the flash memory
15 medium.

16

17 **20. (Cancelled).**

18

19 **21. (Original)** The flash driver as recited in Claim 16, wherein the
20 universal requirements include handling recovery after a power-failure.

21

22 **22. (Original)** The flash driver as recited in Claim 16, wherein the
23 flash medium logic comprises a set of programmable entry points that can be
24 implemented by a user to interface with the type of flash memory medium
25 selected.

1 **23. (Currently Amended)** A processing device that uses a flash
2 memory medium for storage of data, comprising:

3 a file system, configured to control data storage for the processing device;
4 flash media logic, configured to perform physical sector operations to a
5 flash memory medium based on physical sector commands, wherein the flash
6 medium logic comprises a set of programmable entry points that can be
7 implemented by a user to interface with any type of flash memory medium
8 selected, wherein the flash media logic is programmable to permit users to match
9 particular flash medium requirements of a specific manufacturer; and

10 flash abstraction logic, configured to maintain flash memory requirements
11 that are necessary to operate the flash memory medium, wherein the flash memory
12 requirements include managing wear-leveling operations associated with the flash
13 memory medium by way of circular and continuous advancement of a write
14 pointer, wherein the flash media logic and the flash abstraction logic reside within
15 an operating system of the processing device.

16

17 **24. (Original)** The processing device as recited in Claim 23, wherein
18 the flash abstraction logic passes physical logic commands associated with certain
19 types of flash memory medium directly to the flash memory medium logic for
20 translation and execution.

21

22 **25. (Original)** The processing device as recited in Claim 23, wherein
23 the flash abstraction logic is an interface between the flash medium logic and the
24 file system.

1 **26. (Original)** The processing device as recited in Claim 23, wherein
2 the flash memory requirements include maintaining data integrity of the flash
3 memory medium.

4

5 **27. (Cancelled).**

6

7 **28. (Original)** The processing device as recited in Claim 23, wherein
8 the flash memory requirements include handling recovery after a power-failure.

9

10 **29. (Original)** The processing device as recited in Claim 23, wherein
11 the requirements are common to a plurality of different flash memory media.

12

13 **30. (Original)** The processing device as recited in Claim 23, wherein
14 the flash medium logic comprises a set of programmable entry points that can be
15 implemented by a user to perform error code correction with the type of flash
16 memory medium used in the processing device.

17

18 **31. (Original)** The processing device as recited in Claim 23, whereby
19 the flash medium logic relieves the flash abstraction logic from performing
20 translation of the physical sector commands received from the file system.

21

22 **32. (Original)** The processing device as recited in Claim 23, wherein
23 the physical sector operations include read, write and error code correction
24 commands associated with the flash memory medium.

1 **33. (Currently Amended)** In a processing device that uses a flash
2 memory medium for storage of data, a method for driving the flash memory
3 medium, comprising:

4 managing rules associated with operating the flash memory medium in a
5 flash abstraction logic; and

6 issuing physical sector commands directly to the flash memory medium
7 from a flash medium logic, wherein the flash medium logic is programmable to
8 permit users to match particular flash medium requirements of a specific
9 manufacturer;

10 wherein the method is flash memory medium agnostic, and wherein one of
11 the rules includes managing wear-leveling operations associated with the flash
12 memory medium by way of circular and continuous advancement of a write
13 pointer, and wherein the method is performed by way of a component residing
14 within an operating system of the processing device.

15
16 **34. (Original)** The method as recited in Claim 33, wherein one of the
17 rules includes maintaining data integrity of the flash memory medium.

18
19 **35. (Cancelled).**

20
21 **36. (Original)** The method as recited in Claim 33, wherein one of the
22 rules includes handling recovery of the media after a power-failure.

23
24 **37. (Original)** The method as recited in Claim 33, wherein issuing
25 physical sector commands directly to the flash memory medium comprises

1 receiving read and write commands from a file system and translating them into
2 the physical sector commands.

3

4 **38. (Original)** The method as recited in Claim 33, further comprising
5 issuing a set of programmable entry points that can be implemented by a user to
6 perform error code correction with the type of flash memory medium used in the
7 processing device.

8

9 **39. (Original)** The method as recited in Claim 33, further comprising
10 issuing a set of programmable entry points that can be optionally selected by a
11 user to interface with the type of flash memory medium used in the processing
12 device.

13

14 **40. (Original)** The method as recited in Claim 33, further comprising
15 receiving read and write commands from a file system.

16

17 **41. (Currently Amended)** One or more [[tangible]] computer-
18 readable storage media comprising computer-executable instructions that, when
19 executed, perform the method as recited in claim 33.

1 **42. (Currently Amended)** A computer storage media for a flash
2 driver, comprising computer-executable instructions that, when executed, direct
3 the flash driver to provide an interface between a file system, selected from one of
4 a plurality of different file systems, and a flash memory medium, selected from
5 one of a plurality of different flash memory media, wherein the flash driver is
6 located as a component within an operating system and is remote from the flash
7 memory medium, and wherein wear-leveling of the flash memory medium is
8 performed by way of circular and continuous advancement of a write pointer, and
9 wherein the flash driver comprises programmable flash medium logic that is
10 programmable to permit users to match particular flash medium requirements of a
11 specific manufacturer.

12
13 **43. (Currently Amended)** A computer storage media for a flash
14 driver, comprising computer-executable instructions that, when executed, direct
15 the flash driver to:

16 provide an interface between a file system, selected from one of a plurality
17 of different files systems, and a flash memory medium, selected from one of a
18 plurality of different flash memory media; and

19 manage a set of characteristics that are common to the plurality of different
20 flash memory media at a flash abstraction logic;

21 wherein the flash driver is flash memory medium agnostic, and wherein
22 wear-leveling of the flash memory medium is performed by way of circular and
23 continuous advancement of a write pointer, and wherein the flash driver resides as
24 a component within an operating system, wherein the instructions provide

1 programmable flash medium logic that is programmable to permit users to match
2 particular flash medium requirements of a specific manufacturer.

3

4 **44. (Currently Amended)** A computer storage media for a flash
5 driver, comprising computer-executable instructions that, when executed, direct
6 the flash driver to:

7 provide an interface between a file system, selected from one of a plurality
8 of different files systems, and a flash memory medium, selected from one of a
9 plurality of different flash memory media;

10 manage a set of characteristics that are common to the plurality of different
11 flash memory media at a flash abstraction logic; and

12 provide programmable entry points that can be optionally selected by a user
13 to interface with the type of flash memory medium selected;

14 wherein the flash driver is located as a component within an operating
15 system and is remote from the flash memory medium and the flash driver is flash
16 memory medium agnostic, and wherein wear-leveling of a flash memory medium
17 is performed by way of circular and continuous advancement of a write pointer,
18 wherein the instructions provide flash medium logic that is programmable to
19 permit users to match particular flash medium requirements of a specific
20 manufacturer.

1 **45. (Previously Presented)** A method, comprising:
2 providing a processor-executable application, a flash driver residing as a
3 component within the processor-executable application;
4 managing rules associated with operating a flash memory medium by way
5 of the flash driver; and
6 issuing physical sector commands directly to the flash memory medium by
7 way of the flash driver, wherein the method is flash memory agnostic by virtue of
8 the flash driver.

9

10 **46. (Previously Presented)** The method as recited in Claim 45, wherein
11 one of the rules includes maintaining data integrity of the flash memory medium.

12

13 **47. (Previously Presented)** The method as recited in Claim 45,
14 wherein one of the rules includes handling recovery of the media after a power-
15 failure.

16

17 **48. (Previously Presented)** The method as recited in Claim 45,
18 wherein issuing physical sector commands directly to the flash memory medium
19 comprises receiving read and write commands from a file system and translating
20 them into the physical sector commands.

21

22 **49. (Previously Presented)** The method as recited in Claim 45,
23 further comprising issuing a set of programmable entry points that can be
24 implemented by a user to perform error code correction in accordance with the
25 type of flash memory medium used in a processing device.

1 **50. (Previously Presented)** The method as recited in Claim 45,
2 further comprising issuing a set of programmable entry points that can be
3 optionally selected by a user to interface with the type of flash memory medium
4 used in a processing device.

5
6 **51. (Previously Presented)** The method as recited in Claim 45,
7 further comprising wear-leveling of the flash memory medium performed by way
8 of circular and continuous advancement of a write pointer.

9
10 **52. (Previously Presented)** One or more computer storage media
11 comprising computer-executable instructions that, when executed, perform the
12 method as recited in claim 45.

13
14
15
16
17
18
19
20
21
22
23
24
25